PSis - Programação de Sistemas - 2016/2017
Exame 1ª Época, 14 de Junho de 2017, 18h30, Duração: 2h

Software Architectures

1. [1.5 points] Describe two advantages of following a specific Architectural pattern (or style) in the resolution of a problem.

2. [1.5 points] In the project you implemented two different **deployment architecture patterns**.
Describe those two architectural patterns in the context of the project

3. [1.0 point] Explain what is data abstraction and present how it it implemented by the sockets API with respect to management of addresses (UNIX, INET, ...).

Operating Systems

4. [1.0 points] Describe on what way the OS layered architecture increases the security of the Operating System.

5. [1.5 points] The **sendfile** system call allows the transfer of data from a file (fd_read) into another file (fd_write) replacing a **read** and a **write**, as presented in the following examples:

`sendfile(f_write, f_read, NULL, 100);`	`Chat v[100];`
	`read(f_read, v, 100);`
	`write(f_write, v, 100)`

Explain why the use of the **sendfile** system call will be faster than calling the read and the write (as presented before).

Processes/ Shared memory /threads

6. [1.5 point] Explain what happens (to the process terminating, to the Kernel and to the parent process) when the **exit** system call is executed..

7. [1.0 point] What is the mechanism that allows the initial transfer of information from a parent process to is child.

8. [1.0 point] Describe why it is impossible to share linked lists between two **processes** using shared memory.

9. [1.0 point] What is the **difference between threads and processes** that allow linked lists to be shared between threads in the same process.

Process Communication

10. [1.0 point] Describe the main problem of transmitting directly C structures, when implementing communication between multiple processes in different machines.

Sincronização

11. [2 points] In the project, the gateway managed a linked list of peers (searchin, inserting and removing). When a client connected to it, the gateway would find a suitable peer and send its address back to the client:

```
mutex_lock(mux_list);
Peer = search_peer(peer_list);
mutex_unlock(mux_list);
if(peer != NULL)
   send_peer_info(sock_client, peer)
else
   …
```

Explain why the previous pseudo-code for that operation is incorrect with respect to synchronization. Change the code in order to correct it and to guarantee that your solution is efficient.

12. [1.0 point] Explain the two main differences between **spin-locks** and **mutexes** with respect to implementation (one difference) and performance (one difference).

13 [2.0 points] Observe the following code (thread A and Thread B) and change it to guarantee that:

- Code_A1 only executes after the completion of Code_B1
- Code_B2 only executes after the completion of Code A1.

Define how many synchronization variables are needed and initialize them correctly.

Thread A	Thread B

`...`
`while (1){`
`  Code_A1();`
`}`
	`...`
	`while(1){`
	`  Code_B1()`
	`  Code_B2();`
	`}`

Software tests

14. [2 points] Suppose we plan to develop a **circular buffer** data structure. This data structure can be manipulated by the following functions:

- **`b = create_buffer(size)`** – Creates a buffer with certain maximum size
- **`error = add_data(b, data, length)`** – Adds data of a certain length to the buffer **b**, if data does not fit, return error
- **`len = retrieve_data(b, data, length)`** – retrieves data up to a certain length from the buffer **b**, return the length of retrieved data

Describe the various unit tests that should be designed/performed to guarantee that the code fulfills the requirements. Suppose you can access the internal data structure of the buffer

15. [1.0 point] When should **regression tests** be performed in the software development process? What errors are found with it?